REMARKS:

Claims 1-55 were presented for examination and were pending in this application. In an Official Action dated November 1, 2006, claims 1-55 were rejected. Applicants thank Examiner for examination of the claims pending in this application and address Examiner's comments below.

Applicants herein amend claims 1, 5, 7, 16, 17, 26, 34, 36, 45, 48 and 53. These changes are believed not to introduce new matter, and their entry is respectfully requested. The claims have been amended to expedite the prosecution of the application in a manner consistent with the Patent Office Business Goals, 65 Fed. Reg. 54603 (Sept. 8, 2000). In making these amendments, Applicants have not and do not narrow the scope of the protection to which Applicants consider the claimed invention to be entitled and do not concede that the subject matter of such claims was in fact disclosed or taught by the cited prior art. Rather, Applicants reserve the right to pursue such protection at a later point in time and merely seek to pursue protection for the subject matter presented in this submission.

Based on the above Amendment and the following Remarks, Applicants respectfully request that Examiner reconsider all outstanding objections and rejections, and withdraw them.

Objections to the Claims

In the 5th through the 10th paragraphs of the Office Action, Examiner has objected to claims 1, 5, 7, 17, 36 and 48 because of informalities. Applicants have amended claims 1, 5, 7, 17, 36 and 48 accordingly. Thus, Applicants respectfully submit that all informalities with respect to claims 1, 5, 7, 17, 36 and 48 have been resolved.

In view of these amendments, Applicants respectfully request that Examiner withdraw the objections to claims 1, 5, 7, 17, 36 and 48.

Response to Rejection Under 35 USC 112, First Paragraph

Claims 2-18 and 19-28 are rejected as allegedly failing to comply with the enablement requirement. This rejection is respectfully traversed.

Examiner has stated that the specification lacks material that explains how one could make or use an invention that switches threads between consecutive instruction cycles. Applicants kindly point out that applicants have described throughout the specification how threads can be switched between consecutive instruction cycles without incurring a time penalty. For example, at page 17, lines 9-13, the specification states that:

> Zero-time context switching is the ability to switch between one program context and another without incurring any time penalty. This implies that the context switch occurs between machine instructions. (emphasis added)

Applicants also kindly note that, at page 18, lines 1-7, the specification further states:

> As an instruction is passed down the pipeline, a context number is also passed with it. This context number determines which context registers are used to load the program counter, load register values from or to save register values to. Thus, each pipeline stage is capable of operating in separate contexts. Switching between contexts is simply a matter of using a different context number.

Further, in supporting the rejection of claims 2-18 and 19-28 for lack of enablement, the Office Action provides diagrams and states that "[a]s is known, an instruction cycle (or clock cycle) is a period of time in which a clock oscillates from low to high." However, the Examiner provides no support for this statement. Applicants respectfully disagree with Examiner's statement that it is known that an "instruction cycle" is "the period of time in

which a clock oscillates from low to high." Applicants kindly request that the Examiner provide support for this characterization or withdraws this rejection.

Therefore, for at least these reasons, Applicants respectfully submit that the specification contains "a written description of the invention…in such full, clear, concise and exact terms as to enable any person skilled in the art to which it pertains…to make and use the same." See 35 U.S.C. §112 (emphasis added). Thus, Applicants respectfully submit that claims 2-18 and 19-28 are in condition for allowance and kindly request withdrawal of this rejection.

Response to Rejection Under 35 USC 112, Second Paragraph

In the 15th paragraph of the Office Action, Examiner rejects claims 5-12, 16-19, 26, 34-41, 45 and 53 as allegedly not specifically pointing out and distinctly claiming the subject matter that the Applicants regard as their invention.

Claims 5, 16, 26, 34 and 53 are amended herein to correct the informalities noted by the Examiner. These amendments are made so as to more clearly define the invention, and not to narrow the scope of protection provided by the claims. Based on these amendments, applicants kindly request withdrawal of these rejections and allowance of claims 5-12, 16, 26, 34-41 and 53.

Further, the rejection of claims 17 and 19 is based on the same allegation that it is unclear what Applicants mean by "between consecutive instruction cycles" or "between." As shown above, the claimed subject matter is described in and throughout the specification in "such full, clear, concise and exact terms as to enable any person skilled in the art to which it pertains" to make and use the claimed invention. See 35 U.S.C. §112. Thus, for at least

the reasons set forth above, Applicants respectfully submit that the rejection of claims 17 and 19 are also improper and should be withdrawn.

The Examiner writes, "Claim 45 is unclear because it claims that the predetermined fixed schedule is either a fixed strict schedule (Fig. 7a of applicant's drawings), a semi-flexible strict schedule (Fig. 7b of applicant's drawings), or a loose strict schedule (Fig. 7c of applicant's drawings)." Applicants have amended claim 45 to remove the recitation of "a loose strict schedule."

However, Applicants respectfully request clarification of Examiner's statements regarding "semi-flexible strict schedule." Specifically, Applicants are unclear why Examiner states:

> It is not clear how the schedule of claim 1 is also a semi-flexible schedule because a first thread A is allocated processing time every first number of cycles and a second thread B is allocated processing time every second number of processing cycles where the first and second numbers are the same (every 4 cycles). The other threads in this schedule are not fixed and therefore do not apply.

Applicants request clarification as to Examiner's argument "where the first and second numbers are the same," as there is no mention of equal processing cycles regarding the semi-flexible schedule. Applicants also kindly note that the specification provides at page 20, lines 16-20 for example:

> With reference to Figure 7b, when the schedule utilizes a semi-flexible scheduling technique some of the schedule is fixed and the rest of the available quanta are filled with non-real time (NRT) threads. (emphasis added)

As claim 1, from which claim 45 depends, recites a "a predetermined fixed schedule, said schedule specifying that the first thread should be allocated processing time every first number of cycles and that the second thread should be allocated processing time every second number of cycles, wherein said first number of cycles is not equal to said second

number of cycles," Applicants respectfully submit that amended claim 45 specifically points out and distinctly claims the subject matter that the Applicants regard as their invention. Thus, Applicants kindly request withdrawal of this rejection.

Response to Rejection Under 35 USC 102(e) to Borkenhagen

In the 26th paragraph of the Office Action, Examiner rejects claims 1, 29-33, 42, 43 and 45-47 under 35 USC §102(e) as being anticipated by U.S. Patent No. 6,076,157 to Borkenhagen et al. ("Borkenhagen"). This rejection is respectfully traversed.

Claim 1 recites:

> a hardware thread scheduler for identifying which of said program threads said processor executes and configurable to allocate available processing time of the processor among at least the first and second program threads by causing thread-switching at a fixed time according to a predetermined fixed schedule, said schedule specifying that the first thread should be allocated processing time every first number of cycles and that the second thread should be allocated processing time every second number of cycles, wherein said first number of cycles is not equal to said second number of cycles.

The hardware thread scheduler recited in claim 1 is configurable so the processor switches from one thread to another thread according to a predetermined fixed schedule. This is greatly beneficial as it provides a predictable execution time for threads, allowing each thread to be executed for a predetermined number of instruction cycles as specified in the predetermined fixed schedule.

In contrast, Borkenhagen discloses a system and method for data processing in a multithreaded processor where "[t]he thread switch logic has a time-out register which forces a thread switch when execution of the active thread in the multi-threaded processor exceeds a programmable period of time." Borkenhagen, Abstract. The time-out register in

Borkenhagen is used "to force a thread switch to the dormant thread after some time if no useful processing is being accomplished to prevent the system from hanging." Borkenhagen, col. 14, lines 49-51. This forced thread switch prevents a thread from "spinning in a loop unable to do useful work" because it is unable to acquire ownership of, or access to, a necessary resource. Borkenhagen, col. 14, lines 31-34. Thus, the time-out register in Borkenhagen does not specify the processing time allocated to the first and second threads, but rather specifies a maximum time the first and second threads can be inactive before forcing a thread switch.

In the Office Action, Examiner provides an example scenario where a first thread (thread A) has a timeout value of 4 cycles while a second thread (thread B) has a timeout value of 2 cycles. For purposes of illustration, Applicants will use this example scenario to show operation of the time-out register disclosed in Borkenhagen as follows:

AAA…A----BBBB…BBB--AAAA…A----BB….

In the above example, a "-" indicates that the thread is inactive or not producing useful output. Contrary to the Examiner's explanation, the time-out register of Borkenhagen does not specify that thread A should be allocated processing time every 4 cycles and thread B should be allocated processing time every 2 cycles. Rather, the time-out register specifies that when thread A is inactive for 4 cycles (after executing an undefined number of cycles, i.e., AAA…A), a thread switch is forced and thread B is allocated processing time. Similarly, when thread B is inactive for 2 cycles (after executing for an undetermined number of cycles), a thread switch occurs and thread A is allocated processing time. Moreover, the Borkenhagen system cannot guarantee execution of any given thread for any given number of cycles because there is no predetermined time when inactivity will begin

and whether the inactivity will remain for the specific time-out period. Thus, the time-out register in Borkenhagen only specifies the number of wasted cycles for which a thread can be inactive before forcing a thread-switch, not the number of cycles in which a thread is executing. Therefore, Borkenhagen does not disclose the "predetermined fixed schedule, said schedule specifying that the first thread should be allocated processing time every first number of cycles and that the second thread should be allocated processing time every second number of cycles, wherein said first number of cycles is not equal to said second number of cycles" as recited in claim 1.

Similarly, claim 46 recites:

> switching the processor from the first thread state to the second thread state by coupling the execution pipeline from the first set of data storage devices to the second set of storage devices via the hardware thread selector at a fixed time according to a predetermined fixed execution schedule, said execution schedule specifying that the processor should switch to the first thread state every first number of cycles and that the processor should switch to the second thread state every second number of cycles, wherein said first number of cycles is not equal to said second number of cycles

As discussed above, Borkenhagen discloses using a time-out register to force a thread switch when the currently executing thread is inactive or fails to produce a useful result for a defined number of cycles to prevent a background thread from spinning in a loop. Thus, the time-out register in Borkenhagen specifies a maximum number of cycles in which a thread can be inactive or fail to produce useful output before forcing a thread switch. The time-out register in Borkenhagen allows a thread to continue executing indefinitely and does not switch threads until the executing thread fails to produce useful output or becomes inactive for a defined number of cycles. Thus, Borkenhagen fails to disclose that "the processor should switch to the first thread state every first number of cycles and that the processor

should switch to the second thread state every second number of cycles, wherein said first number of cycles is not equal to said second number of cycles" as recited in claim 46.

As claims 29-32 and 42, 43 and 45 are dependent on claim 1, all arguments advanced above with respect to claim 1 are hereby incorporated so as to apply to claims 29-32, 42, 43 and 45.

As claim 47 is dependent on claim 46, all arguments advanced above with respect to claim 46 are hereby incorporated so as to apply to claim 47.

Applicants respectfully submit that for at least these reasons claims 1, 29-32, 42, 43 and 45-47 are patentably distinguishable over the cited reference. Thus, Applicants kindly request withdrawal of these rejections.

Response to Rejection Under 35 USC 103(a) in View of Joy and Emer

In the 39th paragraph of the Office Action, Examiner rejects claims 2-3, 13, 16-17 and 19-24 under 35 USC § 103(a) as allegedly being unpatentable in view of Joy and U.S. Patent No. 6,493,741 to Emer et al. ("Emer"). This rejection is respectfully traversed.

Claim 17 recites:

> a hardware thread scheduler for identifying which of said program threads said pipelined processor executes and configurable to allocate available processing time of the pipelined processor among at least the first and second program threads according to an execution schedule;
>
> wherein said thread selection hardware in the pipelined processor switches from said first thread state to said second thread state between consecutive instruction cycles in response to the hardware thread scheduler identifying which of said program threads said pipelined processor executes.

Similarly, claim 19 recites:

switching the pipelined processor from executing the first program thread to executing the second program thread between the end of an execution cycle and before the beginning of a next consecutive execution cycle by coupling the execution pipeline from the first set of data storage devices to the second set of storage devices via the hardware thread selector.

Switching threads between consecutive instruction cycles beneficially allows switching between one program context and another without incurring any time penalty. Switching from one thread to another between the end of an execution cycle before the beginning of a next consecutive instruction cycle beneficially allows switching to occur without the loss of any execution cycles.

As the Examiner correctly notes, Joy does not disclose that "said thread selection hardware in the pipelined processor switches form said first thread state to said second thread state between consecutive instruction cycles in response to the hardware thread schedule identifying which of said program threads said pipelined processor executes." Joy discloses "oblivious thread switching" in which the thread executed by the processor is switched every N cycles. Joy, col. 17, lines 1-4. However, this disclosure refers to how often a thread switch occurs, not to the time penalty associated with thread switches.

Joy discloses that "the thread switch logic supports a fast thread switch with a very small delay, for example three cycles or less." Joy, col. 16, lines 61-62. Joy additionally discloses that "[t]he thread switch logic generates the TID signal with a thread switch delay or overhead of one processor cycle." Joy, col. 16, lines 3-5. Thus, Joy discloses a delay associated with thread switching of one to three cycles. Hence, in Joy, a first thread is executed during one instruction cycle, a thread switching operation occurs during a second instruction cycle, and a second thread is not executed until a third instruction cycle at the earliest.

The deficiencies in Joy are not rectified by Emer. In its background section, Emer describes a conventional multithreaded architecture in which threads may be switched every cycle. However, as discussed above, the frequency of thread switching does not disclose the time penalty associated with the switching itself. For example, a processor may execute thread A in one instruction cycle, take three instruction cycles to switch to thread B, and then execute thread B for one instruction cycle before switching back to thread A. Emer simply does not address the time required to switch between threads.

Moreover, Emer explicitly discloses:

> On any given cycle, a processor executes instructions from just one of the threads. *On the next cycle, it switches to a different thread context* and executes instructions from the new thread. (emphasis added)

Emer, col. 1, lines 54-58. Hence, Emer, like Joy, discloses that switching to a different thread may take place during one instruction cycle and like Joy fails to disclose switching "between consecutive instruction cycles" or "between the end of an execution cycle and before the beginning of a next consecutive execution cycle "as recited in claims 17 and 19, respectively.

Further, Emer provides that multithreaded processors "better tolerate long-latency operations, effectively eliminating vertical waste." Emer, col. 1, lines 58-60. However, Emer discloses that "[v]ertical waste occurs when a cycle goes completely unused. This can be caused by a long latency instruction, such as a memory access, that inhibits further instruction issue." Emer, col. 1, lines 42-45. In light of this disclosure, Emer's description of "effectively eliminating vertical waste" includes thread switching that takes one or more instruction cycles but avoids cycles that go completely unused due to long latency instructions. Reducing processor stalls caused by long latency events can significantly

improve processor efficiency without reducing the overhead required for thread switching. For example, reducing the number of cycles that go unused by switching threads when a long latency instruction in one thread would stall execution for 5 cycles can "effectively eliminate" vertical waste by executing a different thread while still requiring one or more cycles for switching threads. While Emer refers to the conventional thread switching technique to reduce stalling, Emer fails to address the time required to perform the actual thread switching. Thus, Emer does not disclose "switching between consecutive instruction cycles."

Moreover, the combination of Joy and Emer does not disclose either the "thread selection hardware in the pipelined processor [switching] from said first thread state to said second thread state between consecutive instruction cycles" of claim 17 or the "switching…between the end of an execution cycle and before the beginning of a next consecutive instruction cycle" of claim 19. By combining the "at least one cycle" thread switch delay described in Joy, with the "effective" elimination of vertical waste of Emer, the teachings of the combined references amount to no more than what Joy discloses, that is, thread switching that requires an overhead of at least one cycle. Thus, the claimed thread switching "between consecutive instruction cycles" or "between the end of an execution cycle and before the beginning of a next consecutive execution cycle" is not disclosed by the cited references, both alone and in combination. Therefore, Applicants submit that claims 17 and 19 are patentable over the cited art and request that the rejection be withdrawn.

As claims 2-4, 13 and 16 are dependent from claim 17, all arguments advanced above with respect to claim 17 are hereby incorporated so as to apply to claims 2-4, 13 and 16.

As claims 20-24 are dependent from claim 19, all arguments advanced above with respect to claim 19 are hereby incorporated so as to apply to claims 20-24.

Accordingly, for at least the reasons set forth above, claims 2-4, 13, 16-17 and 19-24 are patentable over the cited references, both alone and in combination. Thus, Applicants kindly request withdrawal of these rejections.

Response to Rejection Under 35 USC 103(a) in View of Joy and Emer in Further View of Ramakrishnan

In the 52nd paragraph of the Office Action, Examiner rejects claims 5-12, 18 and 25-28 under 25 USC § 103(a) as allegedly being unpatentable over Joy in view of Emer and in further view of U.S. Patent No. 6,085,215 to Ramakrishnan et al. ("Ramakrishnan"). This rejection is respectfully traversed.

As claims 5-12 and 18 are dependent on claim 17, all arguments advanced above with respect to claim 17 are hereby incorporated so as to apply to claims 5-12 and 18.

Ramakrishnan is cited to make up for Joy and Emer's lack of "thread identifier for identifying at least one hard-real-time (HRT) thread and at least one non-real-time thread" limitation. Ramakrishnan simply describes a software scheduler that uses a round robin approach to thread scheduling using conventional context switching. See Ramakrishnan, col. 9, lines 9-10. The switching in Ramakrishnan, like in Joy and Emer, is conventional context switching involving "an associated overhead in invoking the new thread." (Ramakrishnan, col. 12, lines 61-62, see also, col. 13, lines 6-7 "avoid time consuming context switching"). Ramakrishnan does not remedy the deficiencies of Joy and Emer, both alone and in

combination, as discussed above. Hence, the combination of Joy, Emer and Ramakrishnan still fails to disclose the "between consecutive instruction cycles" switching recited in claim 17.

Accordingly, for at least the reasons set forth above, claims 5-12, 18 and 25-28 are patentable over the cited references, both alone and in combination. Thus, Applicants kindly request withdrawal of these rejections.

Response to Rejection Under 35 USC 103(a) in View of Joy and Emer in Further View of Borkenhagen

In the 66th paragraph of the Office Action, Examiner rejects claim 14 as allegedly being unpatentable over Joy in view of Emer and in further view of Borkenhagen. This rejection is respectfully traversed.

As claim 14 is dependent on claim 17, all arguments advanced above with respect to claim 17 are hereby incorporated so as to apply to claim 14.

Borkenhagen is cited to make up for the deficiency of "storing said second thread state of said processor during execution of said first program thread" in Joy and Emer. However, the combination of Joy, Emer and Borkenhagen still fails to teach or suggest the "between consecutive instruction cycles" switching recited in claim 17. The system disclosed in Borkenhagen also incurs the conventional "latency and performance penalties associated with switching threads." Borkenhagen, col. 15, lines 37-38. Borkenhagen explicitly discloses:

> In the multithreaded processor in the preferred embodiment described herein, this latency includes the time required to complete execution of

> the current thread to a point where it can be interrupted and correctly restarted when it is next invoked, the time required to switch the thread-specific hardware facilities from the current thread's state to the new thread's state, and the time required to restart the new thread and begin its execution.

Borkenhagen col. 15, lines 38-46. Thus, Borkenhagen also fails to disclose "the pipelined processor switches from said first state to said second thread state between consecutive instruction cycles," so Borkenhagen fails to cure the deficiencies of the Joy and Emer combination.

Accordingly, for at least the reasons set forth above, claim 14 is patentable over the cited references, both alone and in combination. Thus, Applicants kindly request withdrawal of this rejection.

Response to Rejection Under 35 USC 103(a) in View of Joy and Emer in Further View of Levy

In the 68th paragraph of the Office Action, Examiner rejects claim 15 as allegedly being unpatentable over Joy in view of Emer and in further view of U.S. Patent No. 6,314,511 to Levy et al. ("Levy"). This rejection is respectfully traversed.

As claim 15 is dependent on claim 17, all arguments advanced above with respect to claim 17 are hereby incorporated so as to apply to claim 44.

Levy is cited to make up for the combination of Joy and Emer's lack of "said first set of data storage devices comprises registers shared by a plurality of threads." However, Levy discloses a method "for freeing a renaming register, the renaming register being allocated to an architectural register by a processor for the out-of-order execution of at least one of a plurality of instructions." Levy, col. 3, lines 27-30. Levy makes no mention of the time

necessary for switching between threads, but merely discloses a configuration for a "processor with dynamic out-of-order instruction processing capability." Levy, col. 7, lines 18-19. As Levy does not disclose "the pipelined processor switches from said first thread state to said second thread state between consecutive instruction cycles," it fails to remedy the deficient disclosure of the Joy and Emer combination.

Accordingly, for at least the reasons set forth above, claim 15 is patentable over the cited references, both alone and in combination. Thus, Applicants kindly request withdrawal of this rejection.

Response to Rejection Under 35 USC 103(a) in View of Borkenhagen and Ramakrishnan

In the 70th paragraph of the Office Action, Examiner rejects claims 34-41 and 48-55 as allegedly being unpatentable over Borkenhagen in view of Ramakrishnan. This rejection is respectfully traversed.

As claims 34-41 are dependent on claim 1, all arguments advanced above with respect to claim 1 are hereby incorporated so as to apply to claims 34-41. As claims 48-55 are dependent on claim 46, all arguments advanced above with respect to claim 46 are hereby incorporated so as to apply to claims 48-55.

Ramakrishnan is cited to make up for Borkenhagen's lack of "a thread identifier for identifying at least one hard-real-time (HRT) thread and at least one non-real-time thread" limitation. Ramakrishnan simply describes a software scheduler that uses a round robin approach to thread scheduling using conventional context switching. See Ramakrishnan, col.

9, lines 9-10. The scheduler in Ramakrishnan also fails to disclose "a predetermined fixed schedule" for allocating thread processing time. Hence, Ramakrishnan does not remedy the deficiencies of Borkenhagen.

Accordingly, for at least the reasons set forth above, claims 34-41 and 48-55 are patentable over the cited references, both alone and in combination. Thus, Applicants kindly request withdrawal of these rejections.

Response to Rejection Under 35 USC 103(a) in View of Borkenhagen and Levy

In the 87th paragraph of the Office Action, Examiner rejects claim 44 as allegedly being unpatentable over Borkenhagen in view of Levy. This rejection is respectfully traversed.

As claim 44 is dependent on claim 1, all arguments advanced above with respect to claim 1 are hereby incorporated so as to apply to claim 44.

Levy is cited to make up for Borkenhagen's lack of "said first set of data storage devices comprises registers shared by a plurality of threads." However, Levy discloses a method "for freeing a renaming register, the renaming register being allocated to an architectural register by a processor for the out-of-order execution of at least one of a plurality of instructions." Levy, col. 3, lines 27-30. Levy makes no mention of switching threads according to "a predetermined fixed schedule," but merely discloses a configuration for a "processor with dynamic out-of-order instruction processing capability." Levy, col. 7, lines 18-19. As Levy does not disclose "causing thread-switching at a fixed time according

to a predetermined fixed schedule," it fails to remedy the deficient disclosure of Borkenhagen.

Accordingly, for at least the reasons set forth above, claim 44 is patentable over the cited references, both alone and in combination. Thus, Applicants kindly request withdrawal of this rejection.

Conclusion

In sum, Applicants respectfully submit that claims 1 through 55, as presented herein, are patentably distinguishable over the cited references. Therefore, Applicants request reconsideration of the basis for the rejections to these claims and request allowance of them.

In addition, Applicants respectfully invite Examiner to contact Applicants' representative at the number provided below if Examiner believes it will help expedite furtherance of this application.

Respectfully Submitted,
NICHOLAS J. KELSEY, ET AL.

Date: March 27, 2007

By: /Brian G. Brannon/

Brian G. Brannon, Registration No. 57,219
FENWICK & WEST LLP
801 California Street
Mountain View, CA 94041
Phone: (650) 335-7610
Fax: (650) 938-5200
E-Mail: bbrannon@fenwick.com

20880/05093/DOCS/1709510.1